20cys281

Os lab work

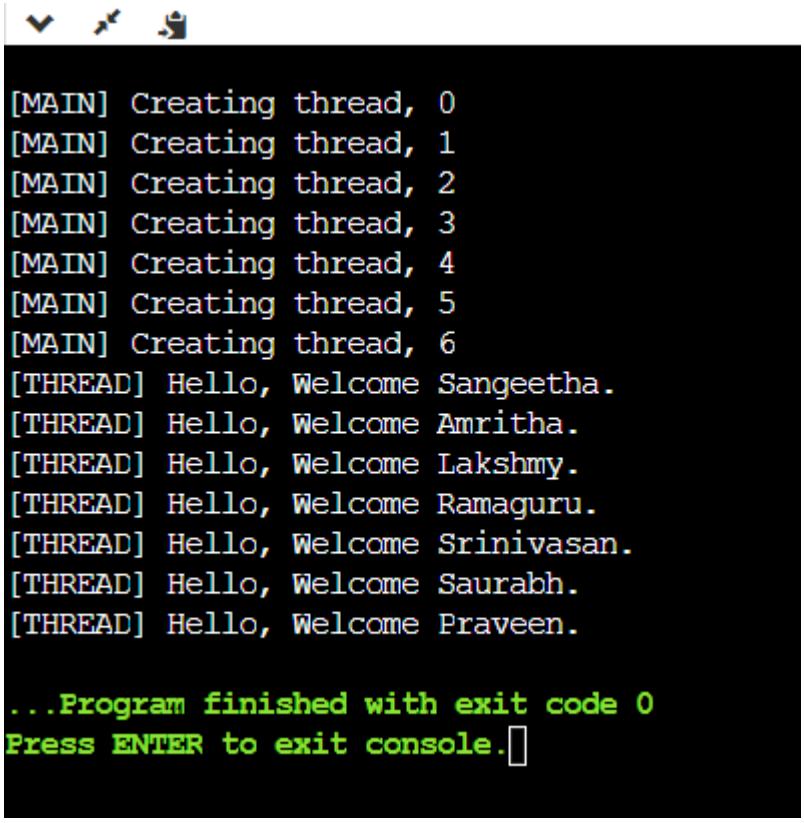
Rakshan.k

cb.en.u4cys21059

1.
```
#include <pthread.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>


void *printWelcomeMessage(void *names) {

   sleep(7);
   char *name = (char *)names;
   printf("\n[THREAD] Hello, Welcome %s.", name);
   pthread_exit(NULL);

}

int main () {


   pthread_t threads[15];
```

```
  char names[10][15] =
{"Amritha","Praveen","Saurabh","Sangeetha","Lakshmy","Srinivasan",
"Ramaguru"};


  int result;


  for(int i = 0; i < 7; i++ ) {


    printf("\n[MAIN] Creating thread, %d", i);



    result = pthread_create(&threads[i], NULL,
printWelcomeMessage, (void *)names[i]);


    if (result) {


      printf("Error in creating thread, %d ", result);
      exit(-1);
    }


  }
```

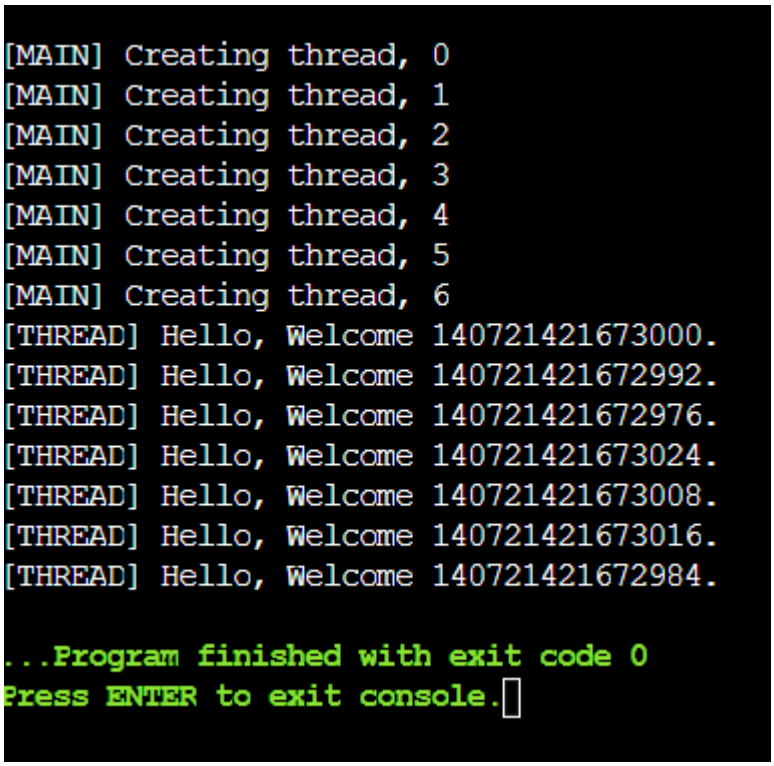
```
    pthread_exit(NULL);
}
```



2.
```
#include <pthread.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>


void *printWelcomeMessage(void *threadid) {

   sleep(10);
```

```
   long tid = (long)threadid;
   printf("\n[THREAD] Hello, Welcome %ld.", tid);
   pthread_exit(NULL);

}

int main () {



   pthread_t threads[7];



   char names[10][15] =
{"Amritha","Praveen","Saurabh","Sangeetha","Lakshmy","Srinivasan",
"Ramaguru"};

   int result;

   for(int i = 0; i < 7; i++ ) {

      printf("\n[MAIN] Creating thread, %d", i);



      result = pthread_create(&threads[i], NULL,
printWelcomeMessage, (void *)&threads[i]);
```

```
    if (result) {

      printf("Error in creating thread, %d ", result);
      exit(-1);
    }

  }


  pthread_exit(NULL);
}
```



3. #include <pthread.h>

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>

struct number {
    int a;
    int b;
};

void *addition ( void *s){

    struct number *z = (struct number *)s;
    printf("Adding %d and %d gives %d\n",z->a,z->b,z->a + z->b);
    pthread_exit(NULL);
}

int main(){

    int result;
    pthread_t threads[5];
    struct number k[5];
    for ( int i = 0; i < 5; i++ ){
```

```
        scanf("%d", &k[i].a);
        scanf("%d", &k[i].b);
        result = pthread_create(&threads[i], NULL, addition,&k[i]);

        if (result) {

         printf("Error in creating thread, %d ", result);
         exit(-1);
      }
    }
    pthread_exit(NULL);
}
```